

**Yee &
Associates, P.C.**

4100 Alpha Road
Suite 1100
Dallas, Texas 75244

Main No. (972) 385-8777
Facsimile (972) 385-7766

**RECEIVED
CENTRAL FAX CENTER**

MAY 27 2005

Facsimile Cover Sheet

To: Commissioner for Patents for Examiner Jacob Lipman Group Art Unit 2134	Facsimile No.: 703/872-9306
From: Jane M. Roberts for Michele Morrow Legal Assistant to Gerald H. Glanzman	No. of Pages Including Cover Sheet: 42
Message: Enclosed herewith: <ul style="list-style-type: none">• Transmittal Document; and• Appeal Brief.	
Re: Application No. 09/751,576 Attorney Docket No: AUS920000797US1	
Date: Friday, May 27, 2005	
Please contact us at (972) 385-8777 if you do not receive all pages indicated above or experience any difficulty in receiving this facsimile.	<i>This Facsimile is intended only for the use of the addressee and, if the addressee is a client or their agent, contains privileged and confidential information. If you are not the intended recipient of this facsimile, you have received this facsimile inadvertently and in error. Any review, dissemination, distribution, or copying is strictly prohibited. If you received this facsimile in error, please notify us by telephone and return the facsimile to us immediately.</i>

**PLEASE CONFIRM RECEIPT OF THIS TRANSMISSION BY
FAXING A CONFIRMATION TO 972-385-7766.**

RECEIVED
MAY 31 2005
OPE/JCW/S

IN THE UNITED STATES PATENT AND TRADEMARK OFFICEIn re application of: **Leung et al.**Serial No.: **09/751,576**Filed: **December 29, 2000****For: Method and Apparatus in a Data
Processing System for a Keystore**§
§
§
§
§
§Group Art Unit: **2134**Examiner: **Jacob Lipman**Attorney Docket No.: **AUS92000797US1****35525**PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

<p>Certificate of Transmission Under 37 C.F.R. § 1.8(a)</p> <p>I hereby certify this correspondence is being transmitted via facsimile to, the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, facsimile number (703) 872-9306 on May 27, 2005.</p> <p>By: <u><i>Jane M. Roberts</i></u> Jane M. Roberts</p>

TRANSMITTAL DOCUMENTCommissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450Sir:
ENCLOSED HEREWITH:

- Appeal Brief (37 C.F.R. 41.37).

A fee of \$500.00 is required for filing an Appeal Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0447. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0447. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0447.

Respectfully submitted,

Gerald H. Glanzman
Gerald H. Glanzman
Registration No. 25,035
Duke W. Yee
Registration No. 34,285
YEE & ASSOCIATES, P.C.
P.O. Box 802333
Dallas, Texas 75380

Docket No. AUS92000797US1

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Leung et al.

Serial No. 09/751,576

Filed: December 29, 2000

For: Method and Apparatus in a Data
Processing System for a Keystore§
§
§
§
§
§
§

Group Art Unit: 2134

Examiner: Lipman, Jacob

RECEIVED
CENTRAL FAX CENTER
MAY 27 2005Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450Certificate of Transmission Under 37C.F.R. § 1.8(a)

I hereby certify this correspondence is being transmitted via facsimile to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, facsimile number (703) 872-9306 on May 27, 2005.

By:


Jane M. Roberts

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on March 29, 2005.

The fees required under § 41.20(B)(2), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

05/31/2005 MBIZUNES 00000032 090447 09751576

01 FC:1402 500.00 DA

(Appeal Brief Page 1 of 40)
Leung et al. - 09/751,576

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-41

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: 3 and 22
2. Claims withdrawn from consideration but not canceled: NONE
3. Claims pending: 1, 2, 4-21 and 23-32
4. Claims allowed: NONE
5. Claims rejected: 1, 2, 4-21 and 23-32
6. Claims objected to: NONE

C. CLAIMS ON APPEAL

The claims on appeal are: 1, 2, 4-21 and 23-32

STATUS OF AMENDMENTS

An Amendment after Final Rejection was not filed. Therefore, claims 1, 2, 4-21 and 23-32 on appeal herein are as presented in the Response to Office Action filed October 15, 2004.

SUMMARY OF CLAIMED SUBJECT MATTER

CLAIM 1 – INDEPENDENT

The subject matter of claim 1 is directed to a method for managing access to data in a keystore in a data processing system. A request for access to an item of data is received from a requestor, (Specification, pg. 15, lines 14 – 16; **Figure 6, step 600**) wherein the item of data is an encrypted item of data, which is encrypted using a first key (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**). A determination of whether the requestor is a trusted requestor is made (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**). This determination is made by checking the requestor's identity against a codebase of trusted users (Specification, pg. 8, lines 26 – 27). In response to a determination that the requestor is a trusted requestor, a copy of the requested item of data is decrypted using a second key (Specification, pg. 8, lines 26 – 27; pg. 10, lines 12-19; **Figure 3, 322; Figure 6, step 606**). The decrypted copy of the requested item of data is sent to the requestor (Specification, pg. 10, lines 19 – 20; pg. 16, lines 4 – 5; **Figure 6, step 608**).

CLAIM 12 – INDEPENDENT

The subject matter of claim 12 is directed to a Keystore system. The Keystore system includes a Keystore object and a Keystore process (Specification pg. 9, line 19 – pg. 10, line 12; **Figure 3, 300, 318**). The Keystore object includes a key and a plurality of entries wherein each entry of the plurality of entries is encrypted using the key (Specification pg. 9, lines 22 – 26; **Figure 3, 302, 310**). The Keystore process provides access to the plurality of entries in response to a request from a trusted application (Specification pg. 10, lines 14 – 15). The determination as to whether or not an application is a trusted application is performed by checking an application's identity against a trusted codebase (Specification, pg. 8, lines 26 – 27). In response to a request from a trusted application, the Keystore process provides the key to the encrypted plurality of entries to the application (Specification pg. 10, lines 17 – 20).

CLAIM 20 – INDEPENDENT

The subject matter of claim 20 is directed to a data processing system for managing access to data in a datastore. The data processing system comprises a receiving means (Specification, pg. 6, lines 8 – 9; **Figure 2, 220**), a determining means (Specification, pg. 6, lines 8 – 9; **Figure 2, 202**), and a sending means (Specification, pg. 6, lines 8 – 9; **Figure 2, 206**). The receiving means receives a request for access to an item of data from a requestor, (Specification, pg. 15, lines 14 – 16; **Figure 6, step 600**) wherein the item of data is an encrypted item of data, which is encrypted using a first key (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**). The determining means determines whether the requestor is a trusted requestor (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**). This determination is made by checking the requestor's identity against a codebase of trusted users (Specification, pg. 8, lines 26 – 27). In response to a determination that the requestor is a trusted requestor, a copy of the requested item of data is decrypted using a second key (Specification, pg. 8, lines 26 – 27; pg. 10, lines 12-19; **Figure 3, 322; Figure 6, step 606**). The sending means sends the decrypted copy of the requested item of data is sent to the requestor (Specification, pg 10, lines 19 – 20; pg. 16, lines 4 – 5; **Figure 6, step 608**).

CLAIM 32 – INDEPENDENT

The subject matter of claim 32 is directed to a computer program product in a computer readable medium for managing access to data in a datastore. A first set of instructions receives a request for access to an item of data from a requestor, (Specification, pg. 15, lines 14 – 16; **Figure 6, step 600**) wherein the item of data is an encrypted item of data, which is encrypted using a first key (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**). A second set of instructions determines whether the requestor is a trusted requestor (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**). This determination is made by checking the requestor's identity against a codebase of trusted users (Specification, pg. 8, lines 26 – 27). In response to a determination that the requestor is a trusted requestor, a third set of instructions decrypts a copy of the requested item of data using a second key (Specification, pg. 8, lines 26 – 27; pg. 10, lines 12-19; **Figure 3, 322; Figure 6, step 606**). A fourth set of instructions sends the decrypted copy of the requested item of data to the requestor (Specification, pg 10, lines 19 – 20; pg. 16, lines 4 –

5; **Figure 6**, step 608).

CLAIM 4 – DEPENDENT

The subject matter of claim 4 is directed to a method for managing access to data in a keystore in a data processing system. A request for access to an item of data is received from a requestor, (Specification, pg. 15, lines 14 – 16; **Figure 6**, step 600) wherein the item of data is an encrypted item of data, which is encrypted using a first key (Specification, pg. 15, lines 16 – 18; **Figure 6**, step 602), and wherein the item of data is another key (Specification, pg. 10, lines 7 – 8; **Figure 3**, 310). A determination of whether the requestor is a trusted requestor is made (Specification, pg. 15, lines 16 – 18; **Figure 6**, step 602). This determination is made by checking the requestor's identity against a codebase of trusted users (Specification, pg. 8, lines 26 – 27). In response to a determination that the requestor is a trusted requestor, a copy of the requested item of data is decrypted using a second key (Specification, pg. 8, lines 26 – 27; pg. 10, lines 12-19; **Figure 3**, 322; **Figure 6**, step 606). The decrypted copy of the requested item of data is sent to the requestor (Specification, pg 10, lines 19 – 20; pg. 16, lines 4 – 5; **Figure 6**, step 608).

CLAIM 6 – DEPENDENT

The subject matter of claim 6 is directed to a method for managing access to data in a keystore in a data processing system. A request for access to an item of data is received from a requestor, (Specification, pg. 15, lines 14 – 16; **Figure 6**, step 600) wherein the item of data is an encrypted item of data, which is encrypted using a first key (Specification, pg. 15, lines 16 – 18; **Figure 6**, step 602), and wherein the item of data is indexed within the Keystore using an alias (Specification, pg. 10, lines 3 – 8; **Figure 3**, 308). A determination of whether the requestor is a trusted requestor is made (Specification, pg. 15, lines 16 – 18; **Figure 6**, step 602). This determination is made by checking the requestor's identity against a codebase of trusted users (Specification, pg. 8, lines 26 – 27). In response to a determination that the requestor is a trusted requestor, a copy of the requested item of data is decrypted using a second key (Specification, pg. 8, lines 26 – 27; pg. 10, lines 12-19; **Figure 3**, 322; **Figure 6**, step 606). The decrypted copy of the requested item of data is sent to the requestor (Specification, pg 10, lines 19 – 20; pg. 16,

lines 4 – 5; **Figure 6**, step 608).

CLAIM 7 – DEPENDENT

The subject matter of claim 7 is directed to a method for managing access to data in a keystore in a data processing system. A request for access to an item of data, wherein the request includes an alias (Specification, pg. 11, lines 16 – 25), is received from a requestor, (Specification, pg. 15, lines 14 – 16; **Figure 6**, step 600) wherein the item of data is an encrypted item of data, which is encrypted using a first key (Specification, pg. 15, lines 16 – 18; **Figure 6**, step 602), and wherein the item of data is indexed within the Keystore using the alias (Specification, pg. 10, lines 3 – 8; **Figure 3**, 308). A determination of whether the requestor is a trusted requestor is made (Specification, pg. 15, lines 16 – 18; **Figure 6**, step 602). This determination is made by checking the requestor's identity against a codebase of trusted users (Specification, pg. 8, lines 26 – 27). In response to a determination that the requestor is a trusted requestor, a copy of the requested item of data is decrypted using a second key (Specification, pg. 8, lines 26 – 27; pg. 10, lines 12-19; **Figure 3**, 322; **Figure 6**, step 606). The decrypted copy of the requested item of data is sent to the requestor (Specification, pg 10, lines 19 – 20; pg. 16, lines 4 – 5; **Figure 6**, step 608). If the requestor is determined not to be a trusted requestor, a null result is sent to the requestor (Specification, pg. 13, lines 19 – 23; pg. 16 lines 8-11; **Figure 6**, step 612).

CLAIM 13 – DEPENDENT

The subject matter of claim 13 is directed to a Keystore system. The Keystore system includes a Keystore object and a Keystore process (Specification pg. 9, line 19 – pg 10, line 12; **Figure 3** 300, 318). The Keystore object includes a key and a plurality of entries wherein each entry of the plurality of entries is encrypted using the key (Specification pg. 9, lines 22 – 26; **Figure 3**, 302, 310), and wherein the plurality of entries is indexed using a plurality of aliases (Specification, pg. 10, lines 3 – 8, 21-23; **Figure 3**, 302, 306). The Keystore process provides access to the plurality of entries in response to a request from a trusted application (Specification pg. 10, lines 14 – 15), wherein the request includes an alias for a requested entry (Specification, pg. 11, lines 16 – 25). The determination as to whether or not an application is a trusted

application is performed by checking an application's identity against a trusted codebase (Specification, pg. 8, lines 26 – 27). In response to a request from a trusted application, the Keystore process provides the key to the encrypted plurality of entries to the application (Specification pg. 10, lines 17 – 20).

CLAIM 14 – DEPENDENT

The subject matter of claim 14 is directed to a Keystore system. The Keystore system includes a Keystore object and a Keystore process (Specification pg. 9, line 19 – pg 10, line 12; **Figure 3 300, 318**). The Keystore object includes a first key, a first plurality of entries wherein each entry of the first plurality of entries is encrypted using the first key (Specification pg. 9, lines 22 – 26; **Figure 3, 302, 310**), and a second plurality of entries corresponding to the first plurality of entries in an unencrypted form encrypted with a second key (Specification, pg. 10, lines 12 – 14; pg. 11, lines 4 – 8; **Figure 3**). The Keystore process provides access to the plurality of entries in response to a request from a trusted application (Specification pg. 10, lines 14 – 15). The determination as to whether or not an application is a trusted application is performed by checking an application's identity against a trusted codebase (Specification, pg. 8, lines 26 – 27). In response to a request from a trusted application, the Keystore process provides the key to the encrypted plurality of entries to the application (Specification pg. 10, lines 17 – 20).14.

CLAIM 23 – DEPENDENT

The subject matter of claim 23 is directed to a data processing system for managing access to data in a datastore. The data processing system comprises a receiving means (Specification, pg. 6, lines 8 – 9; **Figure 2, 220**), a determining means (Specification, pg. 6, lines 8 – 9; **Figure 2, 202**), and a sending means (Specification, pg. 6, lines 8 – 9; **Figure 2, 206**). The receiving means receives a request for access to an item of data from a requestor, (Specification, pg. 15, lines 14 – 16; **Figure 6, step 600**) wherein the item of data is an encrypted item of data, which is encrypted using a first key (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**), and wherein the item of data is another key (Specification, pg. 10, lines 7 – 8; **Figure 3, 310**). The determining means determines whether the requestor is a trusted requestor (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**). This determination is made by checking the requestor's

identity against a codebase of trusted users (Specification, pg. 8, lines 26 – 27). In response to a determination that the requestor is a trusted requestor, a copy of the requested item of data is decrypted using a second key (Specification, pg. 8, lines 26 – 27; pg. 10, lines 12-19; **Figure 3, 322; Figure 6, step 606**). The sending means sends the decrypted copy of the requested item of data is sent to the requestor (Specification, pg 10, lines 19 – 20; pg. 16, lines 4 – 5; **Figure 6, step 608**).

CLAIM 25 – DEPENDENT

The subject matter of claim 25 is directed to a data processing system for managing access to data in a datastore. The data processing system comprises a receiving means (Specification, pg. 6, lines 8 – 9; **Figure 2, 220**), a determining means (Specification, pg. 6, lines 8 – 9; **Figure 2, 202**), and a sending means (Specification, pg. 6, lines 8 – 9; **Figure 2, 206**). The receiving means receives a request for access to an item of data from a requestor, (Specification, pg. 15, lines 14 – 16; **Figure 6, step 600**) wherein the item of data is an encrypted item of data, which is encrypted using a first key (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**), wherein the item of data is indexed within a Keystore using an alias (Specification, pg. 10, lines 3 – 8; **Figure 3, 308**). The determining means determines whether the requestor is a trusted requestor (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**). This determination is made by checking the requestor's identity against a codebase of trusted users (Specification, pg. 8, lines 26 – 27). In response to a determination that the requestor is a trusted requestor, a copy of the requested item of data is decrypted using a second key (Specification, pg. 8, lines 26 – 27; pg. 10, lines 12-19; **Figure 3, 322; Figure 6, step 606**). The sending means sends the decrypted copy of the requested item of data is sent to the requestor (Specification, pg 10, lines 19 – 20; pg. 16, lines 4 – 5; **Figure 6, step 608**).

CLAIM 26 – DEPENDENT

The subject matter of claim 25 is directed to a data processing system for managing access to data in a datastore. The data processing system comprises a receiving means (Specification, pg. 6, lines 8 – 9; **Figure 2, 220**), a determining means (Specification, pg. 6, lines 8 – 9; **Figure 2, 202**), and a sending means (Specification, pg. 6, lines 8 – 9; **Figure 2, 206**). The

receiving means receives a request for access to an item of data from a requestor, (Specification, pg. 15, lines 14 – 16; **Figure 6, step 600**), wherein the request includes an alias (Specification, pg. 11, lines 16 – 25) and wherein the item of data is an encrypted item of data, which is encrypted using a first key (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**), wherein the item of data is indexed within a Keystore using the alias (Specification, pg. 10, lines 3 – 8; **Figure 3, 308**). The determining means determines whether the requestor is a trusted requestor (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**). This determination is made by checking the requestor's identity against a codebase of trusted users (Specification, pg. 8, lines 26 – 27). In response to a determination that the requestor is a trusted requestor, a copy of the requested item of data is decrypted using a second key (Specification, pg. 8, lines 26 – 27; pg. 10, lines 12-19; **Figure 3, 322; Figure 6, step 606**). The sending means sends the decrypted copy of the requested item of data is sent to the requestor (Specification, pg 10, lines 19 – 20; pg. 16, lines 4 – 5; **Figure 6, step 608**). If the requestor is determined not to be a trusted requestor, a returning means sends a null result to the requestor (Specification, pg. 13, lines 19 – 23; pg. 16 lines 8-11; **Figure 6, step 612**).

CLAIM 11 – INDEPENDENT

The subject matter of claim 11 is directed to a method for managing access to data in a keystore in a data processing system. A request for access to an item of data is received from a requestor (Specification, pg. 15, lines 14 – 16; **Figure 6, step 600**), wherein the item of data is an encrypted item of data, which is encrypted using a first key (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**). A determination of whether the requestor is a trusted requestor is made (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**). This determination is made by checking the requestor's identity against a codebase of trusted users (Specification, pg. 8, lines 26 – 27). In response to a determination that the requestor is a trusted requestor, a second key and an encrypted copy of the requested item of data is sent to the requestor (Specification, pg. 3, lines 9 - 11).

CLAIM 15 – INDEPENDENT

The subject matter of claim 15 is directed to a data processing system. The data

processing systems comprises a bus (Specification, pg. 6, lines 2 – 4; **Figure 2, 206**), a communications unit connected to the bus (Specification, pg. 6, lines 14 – 16; **Figure 2, 210**), a memory unit connected to the bus (Specification, pg. 6, lines 8 – 9; **Figure 2, 204**), and a processor unit connected to the bus (Specification, pg. 6, lines 8 – 9; **Figure 2, 202**). Data is sent and received via the communications unit (Specification, pg. 7, lines 25 – 29). A set of instructions is located in the memory (Specification, pg. 7, lines 7 – 11, pg 8, lines 14 – 18). The processor unit executes the set of instructions to receive a request for access to an item of data from a requestor (Specification, pg. 15, lines 14 – 16; **Figure 6, step 600**), wherein the item of data is an encrypted item of data, which is encrypted using a first key (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**). A determination of whether the requestor is a trusted requestor is made (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**). This determination is made by checking the requestor's identity against a codebase of trusted users (Specification, pg. 8, lines 26 – 27). In response to a determination that the requestor is a trusted requestor, a second key and an encrypted copy of the requested item of data is sent to the requestor (Specification, pg. 3, lines 9 - 11).

CLAIM 30 – INDEPENDENT

The subject matter of claim 30 is directed to a data processing system for managing access to data in a datastore. The data processing system comprises a receiving means (Specification, pg. 6, lines 8 – 9; **Figure 2, 220**), a determining means (Specification, pg. 6, lines 8 – 9; **Figure 2, 202**), and a sending means (Specification, pg. 6, lines 8 – 9; **Figure 2, 206**). The receiving means receives a request for access to an item of data is received from a requestor (Specification, pg. 15, lines 14 – 16; **Figure 6, step 600**), wherein the item of data is an encrypted item of data, which is encrypted using a first key (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**). The determining means determines whether the requestor is a trusted requestor (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**). This determination is made by checking the requestor's identity against a codebase of trusted users (Specification, pg. 8, lines 26 – 27). In response to a determination that the requestor is a trusted requestor, the sending means sends a second key and an encrypted copy of the requested item of data is sent to the requestor (Specification, pg. 3, lines 9 - 11).

CLAIM 31 – INDEPENDENT

The subject matter of claim 31 is directed to a computer program product in a computer readable medium for managing access to data in a datastore. A first set of instructions receives a request for access to an item of data from a requestor (Specification, pg. 15, lines 14 – 16; **Figure 6, step 600**), wherein the item of data is an encrypted item of data, which is encrypted using a first key (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**). A second set of instructions determines whether the requestor is a trusted requestor (Specification, pg. 15, lines 16 – 18; **Figure 6, step 602**). This determination is made by checking the requestor's identity against a codebase of trusted users (Specification, pg. 8, lines 26 – 27). In response to a determination that the requestor is a trusted requestor, a third set of instructions sends a second key and an encrypted copy of the requested item of data to the requestor (Specification, pg. 3, lines 9 - 11).

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

GROUND OF REJECTION (Claims 1, 2, 4-21 AND 23-32)

Claims 1, 2, 4-21 and 23-32 stand rejected under 35 U.S.C. § 103(a) as being unpatenable over Cane et al. (U.S. Patent No. 5,940,507) in view of Padgett et al. (U.S. Patent No. 6,167,518).

ARGUMENT

35 U.S.C. § 103(a), Obviousness, Claims 1, 2, 4-21 and 23-32

The examiner has rejected claims 1, 2, 4-21 and 23-32 under 35 U.S.C. § 103(a) as being unpatentable over Cane et al. (U.S. Patent No. 5,940,547) in view of Padgett et al. (U.S. Patent No. 6,167,518).

A. Claims 1, 2, 4-10, 12-14, 20, 21, 23-29 and 32

As to independent claims 1, 12, 20 and 32, the Final Office Action states:

With regard to claims 1, 2, 5, 12, 20, 21, 24, and 32, as best understood, Cane discloses a method for managing access to data in a processing system (column 1 lines 17-19) including, receiving a request for first key (column 3 lines 59-61) encrypted data, determining whether the requestor is trusted, decrypting a copy of the data (column 4 lines 17-19) with a second key (column 4 lines 23-26) and sending the decrypted data (column 4 lines 16-37). Cane discloses using additional security (column 2 lines 32-33), but does not specifically mention identifying the client by checking his identity against a database. Padgett discloses that a key store can be public, as in Cane, but can also be restricted through authentication (column 3 lines 9-15). It would have been obvious for one of ordinary skill in the art to authenticate the client in Cane, for Cane's stated motivation of adding additional security.

Final Office Action dated December 22, 2004.

A fundamental notion of patent law is the concept that invention lies in the new combination of old elements. Therefore, a rule that every invention could be rejected as obvious by merely locating each element of the invention in the prior art and combining the references to formulate an obviousness rejection is inconsistent with the very nature of "invention." Consequently, a rule exists that a combination of references made to establish a *prima facie* case of obviousness must be supported by some teaching, suggestion, or incentive contained in the prior art which would have led one of ordinary skill in the art to make the claimed invention.

The inquiry is not whether each element existed in the prior art, but whether the invention as a whole is obvious in light of the prior art. *Hartness International, Inc. v. Simplicatic Engineering Co.*, 819 F.2d 1100, 2 U.S.P.Q.2d 1826 (Fed. Cir. 1987)

(Appeal Brief Page 16 of 40)
Leung et al. - 09/751,576

The examiner bears the burden of establishing a *prima facie* case of obviousness based on the prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992).

Additionally, in comparing Cane to the claimed invention, the claim limitations of the presently claimed invention may not be ignored in an obviousness determination.

The present invention in claim 1, which is representative of independent claims 12, 20 and 32 with regard to similarly recited subject matter, recites:

1. A method in a data processing system for managing access to data in a keystore, the method comprising:
 - receiving a request for access to an item of data from a requestor, wherein the item of data is encrypted using a first key;
 - determining whether the requestor is a trusted requestor, wherein the determining step is performed by checking a requestor's identity against a trusted codebase;
 - responsive to a determination that the requestor is a trusted requestor, decrypting a copy of the item of data using a second key to form a decrypted item of data; and
 - sending the decrypted item of data to the requestor.

Cane does not teach all the elements as alleged by the Examiner. Specifically, Cane does not teach the feature of "responsive to a determination that the requestor is a trusted requestor, decrypting a copy of the item of data using a second key to form a decrypted item of data." Such a feature is not taught or suggested by Cane. Therefore, claim 1 is not obvious in view of Cane because the features believed to be disclosed by this cited reference are not present.

The Examiner points to column 4, lines 16-37 of Cane, reproduced below for the convenience of the Board, as teaching the feature of "responsive to a determination that the requestor is a trusted requestor, decrypting a copy of the item of data using a second key to form a decrypted item of data."

Upon receipt of the encrypted file 20 and the encrypted key 24, the archive server 30 writes the encrypted file 32 to a magnetic tape 36, or other medium or long term storage which is inexpensive and which need not encompass real time access, via tape drive 34 at step 120. The encrypted key 38 is then written to a tape index file 40 at step 122, thereby associating the magnetic tape volume 36 with the encrypted file 32 and the encrypted key 38. In alternative embodiments, a

further encryption operation may be performed at the archive server on the encrypted file 32 or the encrypted key 38 to add an additional layer of security.

Recovery of a file is accomplished by the archive server referencing the index to obtain the encrypted key and the volume of the encrypted file. The encrypted file is then retrieved from the volume, and both the encrypted key and the encrypted file are transmitted back to the client. The client then recovers the file through the same two stage process used to encrypt. First, the secondary key must be recovered by decrypting the encrypted key with the master. Second, the original file may be recovered by decrypting the encrypted file with the secondary key.

The passage above teaches that an archive server receives an encrypted file and an encrypted key. The archive server stores the encrypted file in a storage medium, and the encrypted key is written to an index file. When a client, who is the owner of the data, wants to access the stored data, the archive server uses the index to obtain the encrypted key and file, and sends them to the client. The client may then use the same two stage process used to encrypt the file to recover the file.

The two stage process used to encrypt the file is explained, in general terms, in Cane, column 3, lines 31 through 44:

An archive transaction for a file stored at the source system encompasses encryption of the file on the source system using a secondary key, encryption of the secondary key on the source system using a master key, and transmission of the encrypted file and the associated encrypted key to the archive server. Transmission is electronic via computer network, or in alternative embodiments by physical delivery of a suitable magnetic medium. The archive server then stores the encrypted file on magnetic tape or another medium of long term storage, and stores the encrypted key along with an index to the tape containing the encrypted file. The master key used to encrypt the secondary key is retained on the source system.

The two stage process is explained in greater detail in Cane, column 3, lines 56 through column 4, line 1:

A key generator 16 then generates a secondary key 18 as shown in step 102, and uses this key to encrypt the file 10 as shown in step 104 to produce an encrypted file 20, at step 106. The master encryption key 22 is then obtained in step 108 and used to encrypt the secondary key in 18, as shown at step 110, and

produce an encrypted key 24, as indicated in step 112. Note that since the same master key is used to encrypt multiple secondary keys it need be generated only once and then reused for successive secondary keys. The encrypted file 20 and encrypted key 24 are then transmitted to the archive server at steps 116 and 118, respectively, while the master key 22 is retained at the source system 8 at step 114.

The two passages cited above teach a system wherein a file that is encrypted is stored at an archive server. The file is encrypted at the source location using a key called a secondary key. The secondary key is then itself encrypted using a key called a master key. The encrypted file and the encrypted secondary key are sent to the archive server. The encrypted file is stored on a long term storage device. The encrypted secondary key along with an index containing the location of the encrypted file are stored together. The master key, which is the key used to encrypt the secondary key is retained at the source system. The master key is not sent to the archive server and the archive server has no control over the master key.

Therefore, to recover an encrypted file, the client, who is the owner of the data, requests the encrypted file and the encrypted key be returned from the archive server. The archive server then retrieves the stored encrypted secondary key and the index containing the location of the encrypted file. The encrypted file is then retrieved. Then the encrypted file and the encrypted secondary key are sent to the client. The requestor has to use the master key to decrypt the encrypted secondary key. Once the secondary key has been decrypted, the secondary key is used. A secondary key is recovered by decrypting the encrypted key with the master key which is retained at the client, and the original file is recovered by decrypting the encrypted file with the secondary key. Therefore, in Cane, only the owner of the data, the possessor of the master key, may actually decrypt the encrypted file and gain access to the stored data. In contradistinction, the present invention receives a request for a first item of data encrypted with a first key and instead, decrypts a copy of the first item of data, using a second key. In other words, the encrypted file in Cane is not a copy of an item of data wherein the original item of data is encrypted using a first key and the copy is encrypted with a second, different key, as in claim 1.

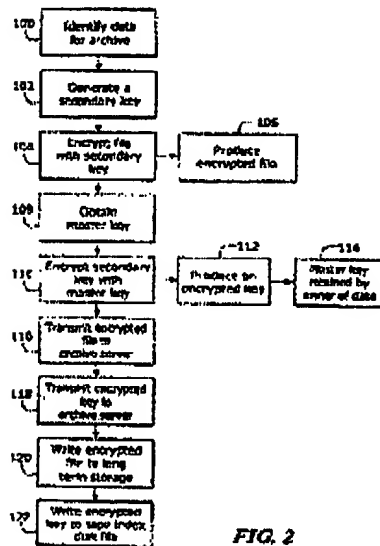
The present invention recites the feature of "responsive to a determination that the requestor is a trusted requestor, decrypting a copy of the item of data using a second key to form a decrypted item of data." According to an aspect of the present invention, when a user adds an

item of data or file to a keystore, a duplicate copy of that item is also created in the keystore. The original item is encrypted using a key, the first key, as normal. However, the duplicate copy of the item is encrypted using a different key, the second key. Therefore, under the method of the present invention when a user adds an item of data, or file, to a keystore, or archive server, the end result is that two version of the item are stored, the original item and a duplicate copy of the item. Each version is separately encrypted using a different key for each item. When a request for the particular original item of data is received by the keystore, the keystore then determines whether or not the requestor belongs to a class of requestors called "trusted requestors." If the requestor is a trusted requestor, the keystore retrieves the encrypted copy of the item of data and uses the second key to decrypt the encrypted copy of the item of data. The keystore sends this decrypted copy of the item of data to the trusted requestor. Therefore, when it has been determined by the keystore that a requestor is a trusted requestor, the keystore decrypts a copy of the item of data using the second key. Cane does not teach or suggest this. Instead, Cane teaches that only one version of the file exists at the archive server, the original encrypted file. Therefore, as Cane teaches that there is only one version of the encrypted file exists at the archive server, Cane cannot teach "decrypting a copy of the item of data using a second key to form a decrypted item of data." Thus, Cane does not teach the feature of "responsive to a determination that the requestor is a trusted requestor, decrypting a copy of the item of data using a second key to form a decrypted item of data."

Additionally, Cane teaches that only a requestor who possesses the master key can decrypt the original encrypted file, since the master key is need to decrypt the encrypted secondary key and the decrypted secondary key is needed to decrypt the encrypted file. Cane in column 3, line 64 to column 4, line 1, further teaches that the master key is retained by the client at the client's own system and it is not sent to the archive server:

The encrypted file 20 and encrypted key 24 are then transmitted to the archive server at steps 116 and 118, respectively, while the master key 22 is retained at the source system 8 at step 114.

Block 114 in Figure 2 further illustrates that the master key is retained at the client in Cane:



Thus, as the master key is needed to decrypt the stored data, and as the master key is stored at the data source/client, *Cane* merely teaches that the source client is the only requestor able to decrypt the stored data. Therefore, *Cane* teaches that the archive server cannot decrypt the stored encrypted original file. Thus, *Cane* cannot teach “decrypting a copy of the item of data using a second key to form a decrypted item of data.” Therefore, *Cane* does not teach the feature of “responsive to a determination that the requestor is a trusted requestor, decrypting a copy of the item of data using a second key to form a decrypted item of data.”

Furthermore, *Cane* does not teach a “trusted requestor.” *Canes* teaches that the requestor for the data, or encrypted file, is the client or owner of the encrypted file and that only this requestor, exclusively, can decrypt the files and access the underlying data since a master key is necessary to decrypt the file and only the client who original sent the encrypted file possesses the master key. This exclusivity is a goal of *Cane* as stated in column 2, lines 63 through 67:

One benefit provided by this arrangement is the elimination of access to data by the archive server, therefore providing the source organization with assurances of access control and privacy, while relieving the source organization of archive cataloging and physical storage duties.

In contrast, as stated on page 11, lines 4 through 7 of the specification:

One feature of the present invention is to create not only the entry that the user or application desired, but an additional entry for use by trusted applications.

Therefore, in claim of the present invention, access to the data contained in the stored file is not limited exclusively to that of the source provider. But, rather, access is specifically intended for and granted to another group of users, which is much broader than merely the source provider of the file, called trusted requestors. The trusted requestors do not have access to the data initially intended to be stored, but only to the copy of the data, as explained in the specification, page 8, lines 24 through 27:

This section duplicates the data in a portion of that Keystore, but protects the data with a different password. Entries from this new section are available to any class from a trusted codebase.

Thus, a client, or source of the encrypted file, as taught by Cane is not the same as a trusted requestor in claim of the present invention. Therefore, Cane does not teach the feature of "responsive to a determination that the requestor is a trusted requestor, decrypting a copy of the item of data using a second key to form a decrypted item of data."

Therefore, for all the reasons stated above, Cane does not teach the feature of "responsive to a determination that the requestor is a trusted requestor, decrypting a copy of the item of data using a second key to form a decrypted item of data." Therefore, claim 1 is not obvious in view of Cane because the features believed to be disclosed by this cited reference are not present. Additionally, Padgett does not cure the deficiencies of Cane. Padgett is cited for the purpose of teaching the feature of "wherein the determining step is performed by checking a requestor's identity against a trusted codebase," and does not teach the feature of "responsive to a determination that the requestor is a trusted requestor, decrypting a copy of the item of data using a second key to form a decrypted item of data." Thus, the combination of Cane with the Padgett reference would not reach the presently claimed invention as recited in claim 1. Therefore, the examiner has failed to state a *prima facie* case of obviousness.

In addition, claim 1 recites the feature of "sending the decrypted item of data to the requestor." This feature is not taught or suggested by Cane. The examiner alleges that this feature

is taught by Cane, column 4, lines 16 through 37, cited above. As was discussed above, and illustrated in Figure 2, reproduced above, Cane teaches that "both the encrypted key and the encrypted file are transmitted back to the client." (Cane, col. 4, lines 31-32). Furthermore, as Cane teaches that a master key, which is retained by the source provider at the source provider's location, is needed to decrypt the encrypted secondary key and that the decrypted secondary key is needed to decrypt the encrypted file, it follows that Cane cannot teach sending a decrypted item of data to the source file as the archive server has no way of decrypting the encrypted file. Therefore, Cane does not teach the feature of "sending the decrypted item of data to the requestor," as recited in claim 1 of the present invention. Therefore, claim 1 is not obvious in view of Cane because the features believed to be disclosed by this cited reference are not present.

Additionally, claim 1 recites the feature of "determining whether the requestor is a trusted requestor, wherein the determining step is performed by checking a requestor's identity against a trusted codebase." The examiner has stated and Appellants agree that Cane does not teach the feature of "wherein the determining step is performed by checking a requestor's identity against a trusted codebase." However, Cane also does not teach "determining whether the requestor is a trusted requestor," as alleged by the examiner. The examiner has consistently mischaracterized this feature as stating "determining whether the requestor is trusted," and has pointed to Cane, column 4, lines 17-19, cited above, as teaching this feature. Cane, column 4, lines 17-19 does not teach this feature. However, another passage of Cane cited by the examiner, column 2, lines 32-33, does refer, in very general terms, to authentication:

Additional security and authentication measures can also be taken.

It appears that the examiner has equated verifying the authenticity of the requestor's identification with "determining whether the requestor is a trusted requestor." However, as was discussed above, a "trusted requestor" refers to a class or groups of classes of users and applications that are authorized to have access to the duplicate items of data stored in the keystore. Therefore, verifying that a requestor is trusted, or authenticating the requestor's identity, is not the same as "determining whether the requestor is a trusted requestor." For example, a user could request access to a piece of data and the authenticity of the user's identification could be verified but the user may still not be a member of one the classes granted

permission to access the stored copy of the item of data. Thus, the user would be "trusted" as asserted by the examiner but the user is not a "trusted requestor" as that term is used in claim 1. Therefore, Cane does not teach "determining whether the requestor is a trusted requestor."

Additionally, as was discussed above, regarding the feature of "responsive to a determination that the requestor is a trusted requestor, decrypting a copy of the item of data using a second key to form a decrypted item of data," Cane does not teach or suggest a "trusted requestor." As Cane does not teach a "trusted requestor," it follows that Cane cannot teach "determining whether the requestor is a trusted requestor." Therefore, Cane does not teach "determining whether the requestor is a trusted requestor."

Therefore, for all the reasons stated above, Cane does not teach the feature of "determining whether the requestor is a trusted requestor, wherein the determining step is performed by checking a requestor's identity against a trusted codebase." Therefore, claim 1 is not obvious in view of Cane because the features believed to be disclosed by this cited reference are not present.

Furthermore, Cane does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. Cane actually teaches away from the presently claimed invention because Cane teaches that the data is stored using a two stage encryption process, involving a master key and a secondary key, whereby the encrypted file and associated encrypted key are stored at an archive server while the original source retains the master key (see Cane, col. 3, lines 32-44). Therefore, Cane teaches that only the source of the data can decrypt the encrypted data. In contrast, the present invention teaches that the archiving server has the ability to decrypt the information stored therein, and that trusted requestors can access and manipulate a stored copy of the data whether or not the data originated with the trusted requestor. Absent some teaching, suggestion, or incentive to modify Cane in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the Applicants' disclosure as a template to make the necessary changes to reach the claimed invention.

Furthermore, Padgett does not cure the deficiencies of Cane. Padgett does not teach the features missing from Cane including "determining whether the requestor is a trusted requestor, wherein the determining step is performed by checking a requestor's identity against a trusted codebase," and "responsive to a determination that the requestor is a trusted requestor, decrypting

a copy of the item of data using a second key to form a decrypted item of data" and "sending the decrypted item of data to the requestor," nor does the examiner point to any portion of Padgett that teaches these features.

The examiner points to Padgett column 3, lines 9 through 15 as teaching "wherein the determining step is performed by checking a requestor's identity against a trusted codebase":

The certificate is stored in a database and is sent to the registrant. Preferably, the database is public with no restriction as to who may access the stored certificate data. Alternatively, access to the database may be restricted to, for example, employees of a particular corporation or government department, database subscribers, or members of a stock exchange.

The above cited passage teaches that a digital certificate can be stored in a database and that it may be desirable to limit access to the database of stored digital certificates based upon a variety of possible limitations. However, nowhere does this passage or Padgett teach how this to accomplish the limiting of access. Nowhere does this passage or any passage of Padgett teach using a trusted codebase. Therefore, Padgett does not teach "wherein the determining step is performed by checking a requestor's identity against a trusted codebase" Therefore, claim 1 is not obvious because the features believed to be disclosed by Padgett are not present. Thus, the combination of the Cane reference with Padgett would not reach the presently claims invention as recited in claim 1. Therefore, the examiner has failed to state a *prima facie* case of obviousness.

Thus, claims 1, 12, 20 and 32 are patentable over the cited references because combination of the Cane reference with the Padgett reference does not teach or suggest the presently claimed invention. Accordingly, for all the above reasons, Appellants submit that the Final Rejection of claims 1, 12, 20 and 32 is improper, and respectfully request that the Final Rejection be reversed.

Claims 2, 4-10, 13, 14, 21 and 23-29 are dependent claims which depend on independent claims 1, 12, 20, and 32. As Appellants have already demonstrated claims 1, 12, 20, and 32 to be in condition for allowance, Appellants respectfully submit that claims 2, 4-10, 13, 14, 21 and 23-29 are also allowable at least by virtue of their depending from an allowable claim.

Accordingly, for all the above reasons, Appellants submit that the Final Rejection of claims 2, 4-10, 13, 14, 21 and 23-29 is improper, and respectfully request that the Final Rejection be reversed.

A.1 Claims 4 and 23

Additionally the claims recite other features not found in the cited references. For example, claims 4 and 23 recite the feature of "wherein the item of data is another key." This feature is not taught or suggested by Cane. The examiner alleges that Cane discloses that "one of the data items is a key, decrypted by the server key." (Final Office Action, page 2). However, the examiner does not give a complete reference as to where in Cane this feature is taught as the reference given by the examiner only states "column 16-26." As there are not 16 columns in Cane, Appellants assume that 16-26 refers to the line numbers and that a reference to the exact column has been omitted. Appellants assume that that the examiner intended to reference column 4, lines 16-26, cited above, in this rejection. However, Cane, column 4, lines 16-26 do not teach the feature of "wherein the item of data is another key." Instead Cane teaches an encrypted file and an encrypted key are returned to the requestor. The requestor then uses a master key to decrypt the encrypted key and the decrypted key is then used to decrypt the encrypted file. Nowhere does this passage of Cane teach or suggest that encrypted file is another, third key. Cane refers to the file as being data. Therefore Cane does not teach the feature of "wherein the item of data is another key," as recited in claims 4 and 23 of the present invention.

A.2 Claims 6, 7, 25 and 26

Additionally claims 6 and 25 recite the feature of "wherein the item of data is indexed within the Keystore using an alias." The examiner points to Cane, column 4, lines 37 through 41 as teaching this feature:

Referring to FIGS. 1 and 3, for file recovery the archive server searches the tape index disk file 40 at step 200 to lookup the encrypted key 44 and the location of the magnetic tape volume 36.

The above cited passage teaches that the encrypted key and the location of the magnetic tape volume containing the encrypted are stored in an index. However, Cane does not teach that encrypted file is indexed using an alias. Therefore, the passage of Cane does not teach the feature of wherein the item of data is indexed within the Keystore using an alias," as recited in claims 6 and 25 of the present invention.

A.2.A Claims 7 and 26

Furthermore, claims 7 and 26 recite the feature of "responsive to an absence of a determination that the requestor is a trusted requestor, returning a null result to the requestor." Cane does not teach this feature. The examiner points to Cane, column 4, lines 37 through 41, cited above, as teaching this feature. However, as was discussed above in regards to claims 6 and 25, Cane, column 4, lines 37 through 41 teaches that the encrypted key and the location of the magnetic tape volume containing the encrypted are stored in an index. Additionally, as was discussed above in regards to claim 1, Cane does not teach a trusted requested. Therefore, it follows that Cane does not teach the feature of responsive to an absence of a determination that the requestor is a trusted requestor, returning a null result to the requestor," as recited in claims 7 and 26 of the present invention.

A.3 Claim 13

Additionally claim 13 recites the feature of "wherein the plurality of entries is indexed using a plurality of aliases and wherein the request includes an alias for a requested entry." Cane does not teach this feature. The examiner points to Cane, column 4, lines 37 through 41, cited above, as teaching this feature. However, as was discussed above in regards to claims 6 and 25, Cane, column 4, lines 37 through 41 teaches that the encrypted key and the location of the magnetic tape volume containing the encrypted are stored in an index. However, the passage of Cane does not teach that a plurality of entries, or encrypted files, is indexed using a plurality of aliases. Nor does the passage of Cane teach that a request for access to an entry includes an alias

The above cited passage teaches that the encrypted key and the location of the magnetic tape volume containing the encrypted are stored in an index. However, Cane does not teach that encrypted file is indexed using an alias. Therefore, the passage of Cane does not teach the feature of wherein the item of data is indexed within the Keystore using an alias,” as recited in claims 6 and 25 of the present invention.

A.2.A Claims 7 and 26

Furthermore, claims 7 and 26 recite the feature of “responsive to an absence of a determination that the requestor is a trusted requestor, returning a null result to the requestor.” Cane does not teach this feature. The examiner points to Cane, column 4, lines 37 through 41, cited above, as teaching this feature. However, as was discussed above in regards to claims 6 and 25, Cane, column 4, lines 37 through 41 teaches that the encrypted key and the location of the magnetic tape volume containing the encrypted are stored in an index. Additionally, as was discussed above in regards to claim 1, Cane does not teach a trusted requested. Therefore, it follows that Cane does not teach the feature of responsive to an absence of a determination that the requestor is a trusted requestor, returning a null result to the requestor,” as recited in claims 7 and 26 of the present invention.

A.3 Claim 13

Additionally claim 13 recites the feature of “wherein the plurality of entries is indexed using a plurality of aliases and wherein the request includes an alias for a requested entry.” Cane does not teach this feature. The examiner points to Cane, column 4, lines 37 through 41, cited above, as teaching this feature. However, as was discussed above in regards to claims 6 and 25, Cane, column 4, lines 37 through 41 teaches that the encrypted key and the location of the magnetic tape volume containing the encrypted are stored in an index. However, the passage of Cane does not teach that a plurality of entries, or encrypted files, is indexed using a plurality of aliases. Nor does the passage of Cane teach that a request for access to an entry includes an alias

11. A method in a data processing system for managing access to data in a keystore, the method comprising:
- receiving a request for access to an item of data from a requestor, wherein the item of data is encrypted using a first key;
 - determining whether the requestor is a trusted requestor, wherein the determining step is performed by checking a requestor's identity against a trusted codebase; and
 - responsive to a determination that the requestor is a trusted requestor, sending a second key and an encrypted copy of the item of data to the requestor.

As was discussed above in regards to claim 1, Cane does not teach the feature of "determining whether the requestor is a trusted requestor, wherein the determining step is performed by checking a requestor's identity against a trusted codebase." Therefore, claim 11 is not obvious in view of Cane because the features believed to be disclosed by this cited reference are not present. Additionally, as was discussed above in regards to claim 1, Padgett does not cure the deficiencies of Cane. Padgett is cited for the purpose of teaching the feature of "wherein the determining step is performed by checking a requestor's identity against a trusted codebase," which, as was discussed above regarding claims 1, Padgett does not teach. Thus, the combination of Cane with the Padgett reference would not reach the presently claims invention as recited in claim 11. Therefore, the examiner has failed to state a *prima facie* case of obviousness.

Additionally, claim 11 recites the feature of "responsive to a determination that the requestor is a trusted requestor, sending a second key and an encrypted copy of the item of data to the requestor." This feature is not taught or suggested by Cane. The examiner points to Cane, column 4, lines 29 through 31, reproduced above, as teaching this feature. However, Cane, column 4, lines 29 through 31 teaches that "both the encrypted key and the encrypted file are transmitted back to the client." Sending an encrypted key and the original encrypted file to the client is not the same as sending a second, unencrypted key and an encrypted copy of the file. Therefore Cane does not teach the feature of "responsive to a determination that the requestor is a trusted requestor, sending a second key and an encrypted copy of the item of data to the requestor."

Furthermore, as was discussed above regarding claim 1, Cane does not teach a trusted requestor. Therefore Cane cannot teach the feature of "responsive to a determination that the requestor is a trusted requestor, sending a second key and an encrypted copy of the item of data to the requestor." Therefore, claim 11 is not obvious in view of Cane because the features

believed to be disclosed by this cited reference are not present. Thus, the combination of Cane with the Padgett reference would not reach the presently claims invention as recited in claim 11. Therefore, the examiner has failed to state a *prima facie* case of obviousness.

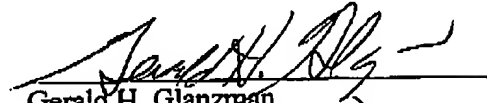
Thus, claims 11, 15, 30 and 31 are patentable over the cited references because combination of the Cane reference with the Padgett reference does not teach or suggest the presently claimed invention. Accordingly, for all the above reasons, Appellants submit that the Final Rejection of claims 11, 15, 30 and 31 is improper, and respectfully request that the Final Rejection be reversed.

Claims 16-19 are dependent claims which depend on independent claim 15. As Appellants have already demonstrated claim 15 to be in condition for allowance, Appellants respectfully submit that claims 16-19 are also allowable at least by virtue of their depending from an allowable claim.

Accordingly, for all the above reasons, Appellants submit that the Final Rejection of claims 16-19 is improper, and respectfully request that the Final Rejection be reversed.

CONCLUSION

For all the above reasons, Appellants submit that the Final Rejection of claims 1, 2, 4-21 and 23-32 is improper, and respectfully request that the Final Rejection be reversed.


Gerald H. Glanzman
Registration No. 25,035
YEE & ASSOCIATES, P.C.
PO Box 802333
Dallas, TX 75380
(972) 385-8777

GHG/bj

CLAIMS APPENDIX

The text of the claims involved in the appeal are:

1. A method in a data processing system for managing access to data in a keystore, the method comprising:
 - receiving a request for access to an item of data from a requestor, wherein the item of data is encrypted using a first key;
 - determining whether the requestor is a trusted requestor, wherein the determining step is performed by checking a requestor's identity against a trusted codebase;
 - responsive to a determination that the requestor is a trusted requestor, decrypting a copy of the item of data using a second key to form a decrypted item of data; and
 - sending the decrypted item of data to the requestor.
2. The method of claim 1, wherein the requestor is an application.
4. The method of claim 1, wherein the item of data is another key.
5. The method of claim 1, wherein the item of data is a certificate.
6. The method of claim 1, wherein the item of data is indexed within the Keystore using an alias.

7. The method claim 6, wherein the request includes the alias further comprising:
responsive to an absence of a determination that the requestor is a trusted requestor,
returning a null result to the requestor.
8. The method of claim 1 further comprising:
responsive to receiving a request to add a new item of data to the Keystore, encrypting the
new item of data to form an encrypted item of data; and
storing the encrypted item of data in the Keystore.
9. The method of claim 8 further comprising:
storing an encrypted copy of the new item of data in the Keystore.
10. The method of claim 8, wherein each item of data in the Keystore is associated with an
alias.
11. A method in a data processing system for managing access to data in a keystore, the
method comprising:
receiving a request for access to an item of data from a requestor, wherein the item of data
is encrypted using a first key;
determining whether the requestor is a trusted requestor, wherein the determining step is
performed by checking a requestor's identity against a trusted codebase; and
responsive to a determination that the requestor is a trusted requestor, sending a second
key and an encrypted copy of the item of data to the requestor.

12. A Keystore system comprising:

a Keystore object including:

a key; and

a plurality of entries, wherein each entry within the plurality of entries is encrypted using the key; and

a Keystore process, wherein the Keystore process provides access to the plurality of entries in response to a request from a trusted application by providing the key to the trusted application and in response to a determination that the application is a trusted application, wherein the determination is performed by checking an application's identity against a trusted codebase.

13. The Keystore system of claim 12, wherein the plurality of entries is indexed using a plurality of aliases and wherein the request includes an alias for a requested entry.

14. The Keystore system of claim 12, wherein the plurality of entries is a first plurality of entries and wherein the Keystore object includes a second plurality of entries corresponding to the first plurality of entries in an unencrypted form encrypted with a second key.

15. A data processing system comprising:

a bus system;

a communications unit connected to the bus, wherein data is sent and received using the communications unit;

a memory connected to the bus system, wherein a set of instructions are located in the memory; and

a processor unit connected to the bus system, wherein the processor unit executes the set of instructions to receive a request for access to an item of data from a requestor, wherein the item of data is encrypted using a first key, determine whether the requestor is a trusted requestor, wherein the determining step is performed by checking a requestor's identity against a trusted codebase, and send a second key and an encrypted copy of the item of data to the requestor in response to a determination that the requestor is a trusted requestor.

16. The data processing system of claim 15, wherein the bus system includes a primary bus and a secondary bus.

17. The data processing system of claim 15, wherein the processor unit includes a single processor.

18. The data processing system of claim 15, wherein the processor unit includes a plurality of processors.

19. The data processing system claim 15, wherein the communications unit is an Ethernet adapter.

20. A data processing system for managing access to data in a datastore, the data processing system comprising:

receiving means for receiving a request for access to an item of data from a requestor, wherein the item of data is encrypted using a first key;

determining means for determining whether the requestor is a trusted requestor, wherein the determining step is performed by checking a requestor's identity against a trusted codebase; and

decrypting means, responsive to a determination that the requestor is a trusted requestor, for decrypting a copy of the item of data using a second key to form a decrypted item of data; and

sending means for sending the decrypted item of data to the requestor.

21. The data processing system of claim 20, wherein the requestor is an application.

23. The data processing system of claim 20, wherein the item of data is another key.

24. The data processing system of claim 20, wherein the item of data is a certificate.

25. The data processing system of claim 20, wherein the item of data is indexed within the Keystore using an alias.

26. The data processing system claim 25, wherein the request includes the alias further comprising:

returning means, responsive to an absence of a determination that the requestor is a

trusted requestor, for returning a null result to the requestor.

27. The data processing system of claim 20 further comprising:

encrypting means, responsive to receiving a request to add a new item of data to the Keystore, for encrypting the new item of data to form an encrypted item of data; and
storing means for storing the encrypted item of data in the Keystore.

28. The data processing system of claim 27, wherein the storing means is a first storing means further comprising:

second storing means for storing the new item of data in the Keystore.

29. The data processing system of claim 27, wherein each item of data in the Keystore is associated with an alias.

30. A data processing system for managing access to data in a datastore, the data processing system comprising:

receiving means for receiving a request for access to an item of data from a requestor, wherein the item of data is encrypted using a first key;

determining means for determining whether the requestor is a trusted requestor, wherein the determining step is performed by checking a requestor's identity against a trusted codebase; and

sending means, responsive to a determination that the requestor is a trusted requestor, for sending a second key and an encrypted copy of the item of data to the requestor.

31. A computer program product in a computer readable medium for managing access to data in a datastore, the computer program product comprising:

first instructions for receiving a request for access to an item of data from a requestor, wherein the item of data is encrypted using a first key;

second instructions for determining whether the requestor is a trusted requestor, wherein the determining step is performed by checking a requestor's identity against a trusted codebase; and

third instructions, responsive to a determination that the requestor is a trusted requestor, for sending a second key and an encrypted copy of the item of data to the requestor.

32. A computer program product in a computer readable medium for managing access to data in a datastore, the computer program product comprising:

first instructions for receiving a request for access to an item of data from a requestor, wherein the item of data is encrypted using a first key;

second instructions for determining whether the requestor is a trusted requestor, wherein the determining step is performed by checking a requestor's identity against a trusted codebase;

third instruction, responsive to a determination that the requestor is a trusted requestor, for decrypting a copy of the item of data using a second key to form a decrypted item of data; and

fourth instructions for sending the decrypted item of data to the requestor.

EVIDENCE APPENDIX

There is no evidence to be presented.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings.